AMENDMENTS TO SPECIFICATION

1. Please change the title of the application to:

Delivering a Graphical Object to a Browser

2. Please replace the paragraph at page 1, lines 6-8 with the following amended paragraph:

A₁

DELIVERING A GRAPHICAL OBJECT TO A BROWSER ~~APPLICATION-~~
~~INDEPENDENT SYSTEM AND PROCESS FOR DYNAMICALLY~~
~~GENERATING LOW-COMPLEXITY GRAPHICS EMBEDDED AS WEB~~
~~CONTENT USING A TAG-DELIMITED SCRIPT~~

3. Please change the paragraph at page 1, lines 10-13 to:

A₂

The present invention relates in general to ~~low-complexity~~ graphics generation and, in
particular, to <u>delivering a graphical object to a browser</u> ~~an application-independent~~
~~system and process for dynamically generating low-complexity graphics embedded as~~
~~Web content using a tag-delimited script~~.

4. Please change the paragraphs at page 2, line 24 – page 3, line 14:

A₃

~~In the prior art, four~~ <u>Three</u> approaches provide graphical Web content to individual Web
browsers. In the first ~~prior art~~ approach, graphical Web content is generated in advance of
download on a Web server. A graphics engine using a graphics toolkit, such as the Java
Abstract Window toolkit (AWT), generates graphics for download and stores the
graphical data for retrieval by Web servers upon client request. This approach requires *a
priori* knowledge of expected graphical Web content and limits the level of user
interactivity to selecting from amongst pre-generated graphics. As well, this approach
also suffers from long download times and is memory intensive.

In the second ~~prior art~~ approach, a graphics engine using a graphical toolkit generates graphical content *ad hoc* rather than ahead of download time. New graphical content is generated responsive to each new user request and requires the download of a large image. This approach imposes long download times and also requires the user to wait out a sometimes time-consuming graphics generation routine upon each request. This approach is usually memory intensive.

A3

In the third ~~prior art~~ approach, Web browser-executable modules, such as applets, are used to render graphical images on a client at the time of request. This approach has the advantage of decreasing download times for individual graphical images, but still requires the download of individual applets prior to the generation of each graphic. Moreover, applets are generally non-portable and application-dependent.

5. Please delete the paragraph at page 3, lines 15-18.

6. Please replace the paragraph at page 4, lines 14-24 with the following amended paragraph:

An embodiment of the present invention provides an ~~applicant~~ <u>application</u>-independent system and process for dynamically generating low-complexity graphics embedded as Web content using a tag-delimited script. A graphics object class defining a logical canvas and including a vector of row objects is specified. Each graphics object class contains a set of cell objects defining display attributes for a uniform rectangular region. A shape is drawn onto the logical canvas by sequentially parsing through each row object in the vector and through each cell object contained therein to consistently structure the logical canvas. The logical canvas is converted into a table encoded in a tag-delimited

A4

script by converting in order each row object into a row within the table, and each cell

object into a cell within each row.

7. Please replace the paragraph at page 9, lines 21-24 with the following amended paragraph:

The individual HTML row objects ~~must not~~ generally do not overlap and [[must]] can be

arranged in numerical order within the vector. The vector is updated as new shapes

containing uniform rectangular regions are drawn. A public constructor method

HtmlGraphics is defined to instantiate a new *HtmlGraphics* graphics object 50.

8. Please replace the paragraph at page 23, lines 1-3 with the following amended paragraph:

DELIVERING A GRAPHICAL OBJECT TO A BROWSER ~~APPLICATION-~~

~~INDEPENDENT SYSTEM AND PROCESS FOR DYNAMICALLY~~

~~GENERATING LOW-COMPLEXITY GRAPHICS EMBEDDED AS WEB~~

~~CONTENT USING A TAG-DELIMITED SCRIPT~~

9. In the Abstract, please replace the paragraph at page 23, lines 5-14, with a new paragraph:

Techniques are provided for delivering a graphical object to a browser including the steps of receiving a request that requires delivery of the graphical object to the browser, wherein the graphical object is not a table; and in response to the request, generating a script which, when executed by the browser, causes the browser to render the graphical object as a particular table.